SPECKLE TOOL V1.2

USER GUIDE

December 2012

Project

Scientific Initiation Project Michel Melo da Silva, undergraduate student in the 9th period of the course Computer Science by Federal University of Lavras, under supervision of professor Phd. Roberto Alves Braga Jr, from Engineering Department.

Funding and Collaborators

Project developed at Centre for Development of Applied Instrumentation in Agriculture, funded by Financier of Studies and Projects Brazilian Organization with support from the Foundation for Research Support of the State of Minas Gerais and scholarship from the National Council for Scientific and Technological Development.

Team

Research team linked to development, Professor Dr. Roberto Alves Braga Jr (robbraga@gmail.com), Michel Melo da Silva (michel.silva66@gmail.com), Robson Pierangeli Godinho (robson.pierangeli@gmail.com), José Roberto de Almeida Nozela (josenozela@gmail.com).

Objective

Help researchers in the Dynamic Laser Speckle area when they are assembling the setup and lighting of their experiments, providing them feedback in real time.

Dates

Project Start: 01/01/2011

Last Actualization: 08/11/2012

Software Register in National Institute of Industrial Property

Registration Protocol: *014110003031*

Registration Number: 12471-6

Type of Registration: *Software free to use.*

Index:

1. Basic Considerations

1.1 Programming Language Utilized

The project was implemented using the programming language C++ with the assistance of Qt Framework for the graphical development of user interface and use of additional basic structures. It was also used computer vision library Open Source Computer Vision (OpenCV) for efficient manipulation of images.

1.2 Generated Files

At the option of video recording the program will save a DIVX encoded video in format ".avi". Therefore it is necessary a codec to execute and to read for that encoding. The program saves images in Bitmap Image format (".bmp").

1.3 Installation and Execution

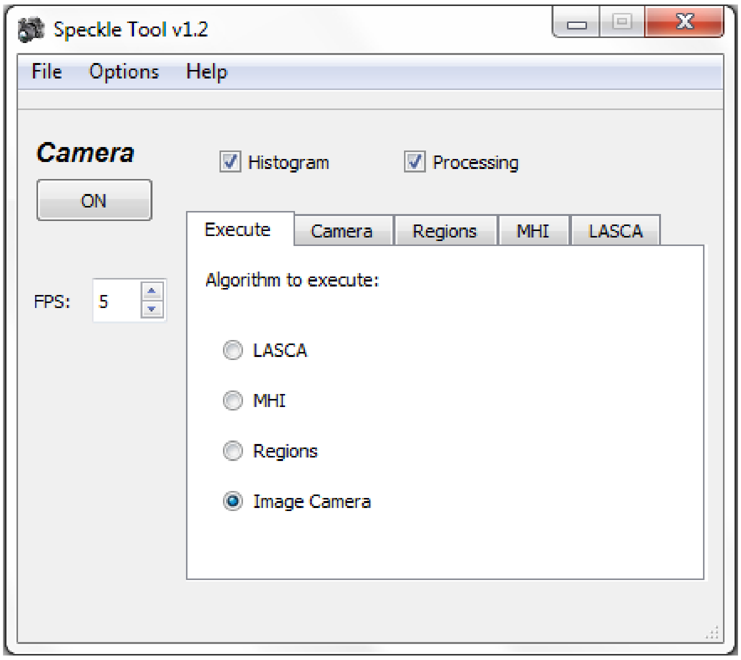
This is an executable program, so it is not necessary to install it. To execute it just copy the folder to your Hard Disk (HD). In this folder there are some files that are essential for the proper functioning of the program, they are: sub-folders named “icons” and “img”, and nine (9) dlls flies called: libcv200.dll, libcxcore.dll, libcc_s_dw2-1.dll, libhighgui200.dll, mingwm10.dll, QtCore4.dll, QtCored4.dl, QtGui4.dll e QtGuid4.dll.

All of these files are already inside the folder. Do **not** remove the program executable file from inside this folder because it will not work anymore. It is possible to create a shortcut for easy access to the executable file.

For the proper functioning of the functions open videos and save videos is necessary to install the "OpenCV-2.0.0a-win32" or higher.

2. Overview

This is the Speckle Tool v1.2 main window.



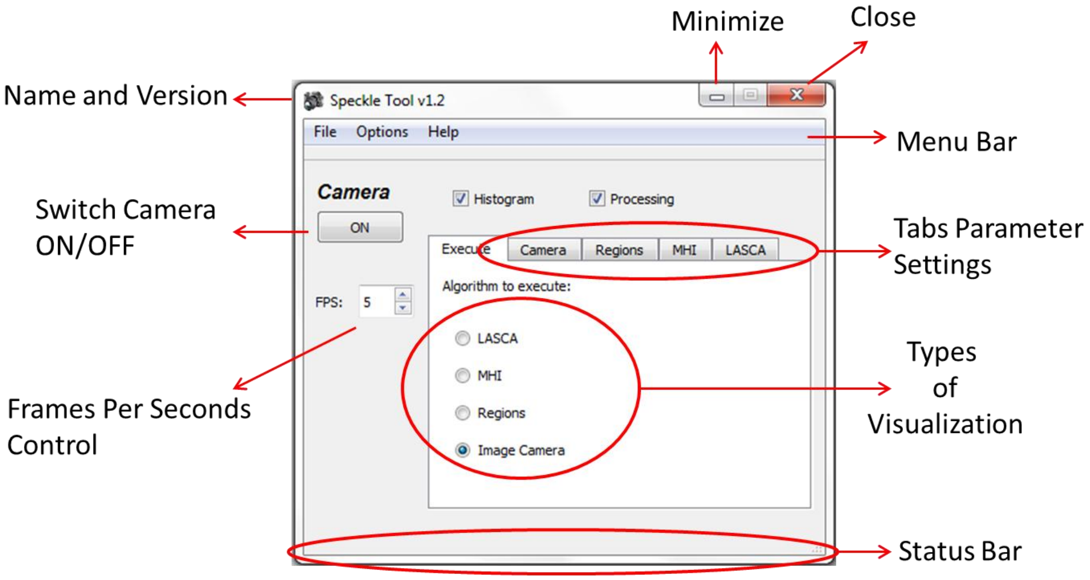
☑ Histogram ☑ Processing

FPS: 5

Algorithm to execute:

○ LASCA
○ MHI
○ Regions
● Image Camera

This window is divided so as to group functions to facilitate interaction with the user, providing a better user experience and usability.



- Name and Version:

At the top left is the name and version of the program you are using, always check the if the manual, documentation and/or information that you have refer to the same version you are using.

- Minimize and Close the Window:

At the top right there are buttons to minimize and close the window. **There is no** button to maximize the window, it cannot be resized.

- Status Bars:

In the status bar will appear indications (feedback) from program to the user, informing the actions are being executed, were finalized and system information.

- Types de Visualization:

At the Types of Visualization selectable fields (selected algorithm), only one item can be selected each time.

- Switch the Camera ON/OFF:

When the "ON" button is pressed, the camera will turn on, the image processing window and the histogram window will appear and that button will be called "OFF". Clicking on the "OFF" button the camera will turn off and the windows opened, and histogram display, will close.

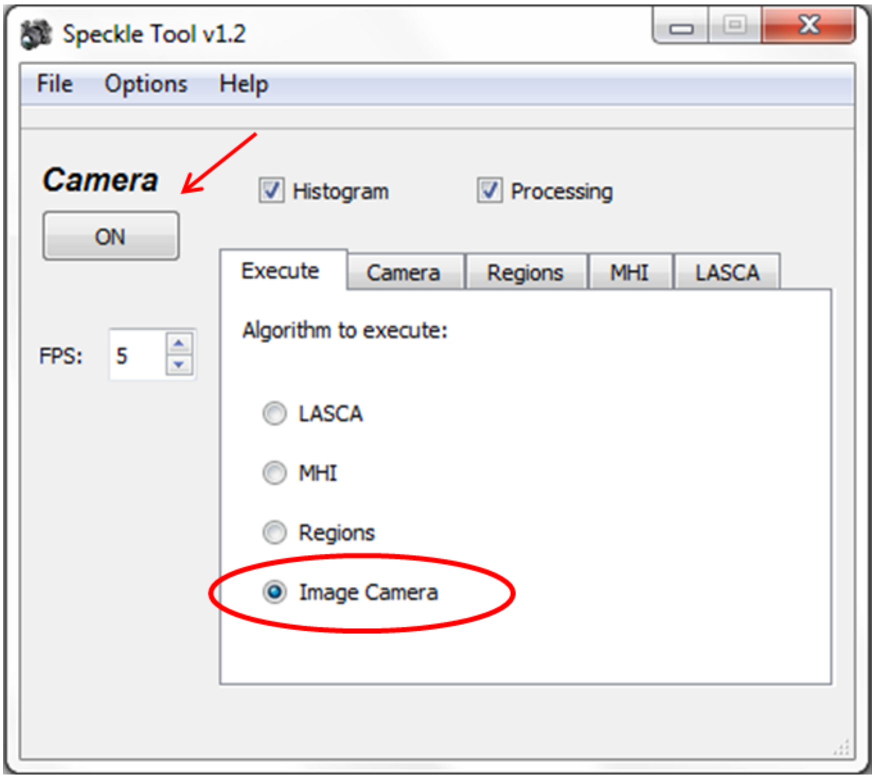
3. Execute Tab

In this Tab are the options to available algorithms for image processing.

3.1. Visualization of Camera Image and Color Processing on this

To view the camera image must select the field "Image Camera" in "Execute" tab on the main program window, and then click in the button ON.

You will be directed to Camera Tab (Chapter 4), where you can change the visualization parameters.



☑ Histogram ☑ Processing

FPS: 5

Algorithm to execute:

○ LASCA

○ MHI

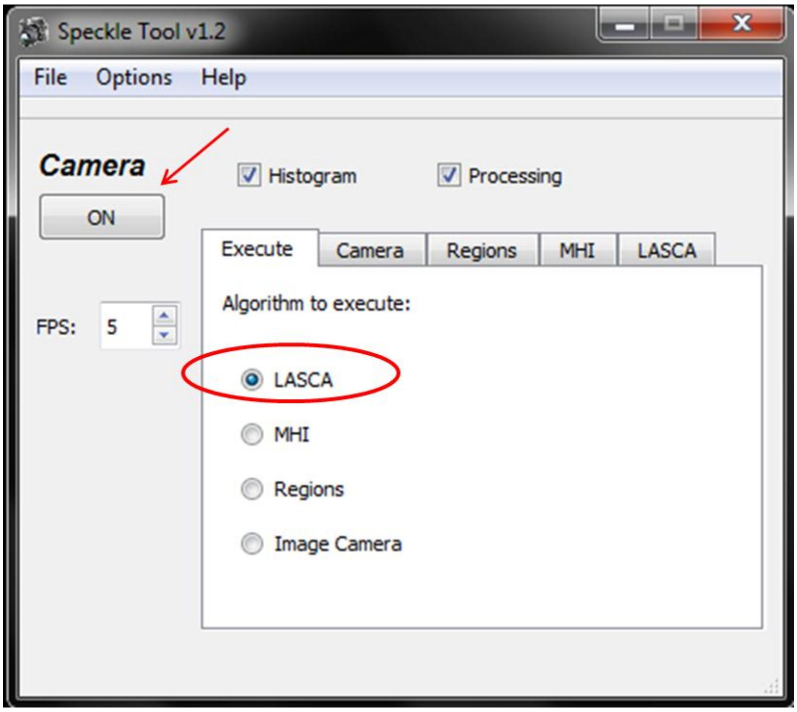
○ Regions

● Image Camera

3.2. Visualization of LASCA Algorithm

To execute the LASCA algorithm on the images must select the field LASCA at the “Execute” tab, on the program main window and then click in the button ON, case the camera is off.

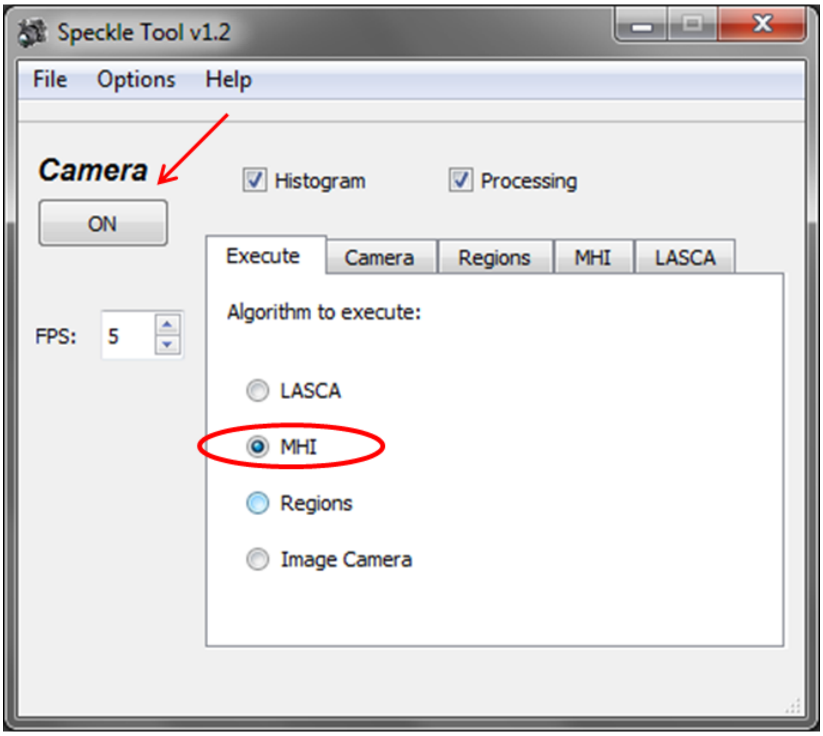
You will be directed to LASCA Tab (Chapter 5) automatically, where you can change the visualization and algorithm parameters.



3.3. Visualization of MHI Algorithm

To execute the MHI processing on the images must select the field MHI at the “Execute” tab, on the program main window and then click in the button ON, case the camera is off.

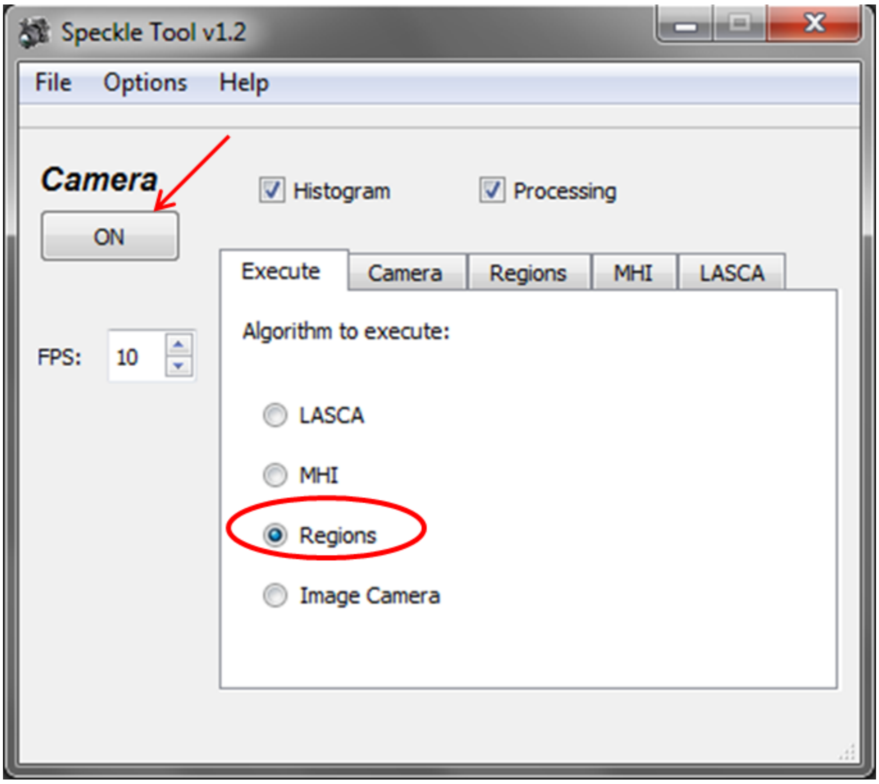
You will be directed to MHI Tab (Chapter 6) automatically, where you can change the visualization and algorithm parameters.



3.4. Visualization of Regions Method

To execute this processing must check the field "Regions" in the "Run" tab, on the program main window and then click in the button "ON" if the camera is off.

You will be directed to the tab "Regions" (Chapter 7) automatically, where you can change the visualization and method parameters.



Algorithm to execute:

○ LASCA

○ MHI

● Regions

○ Image Camera

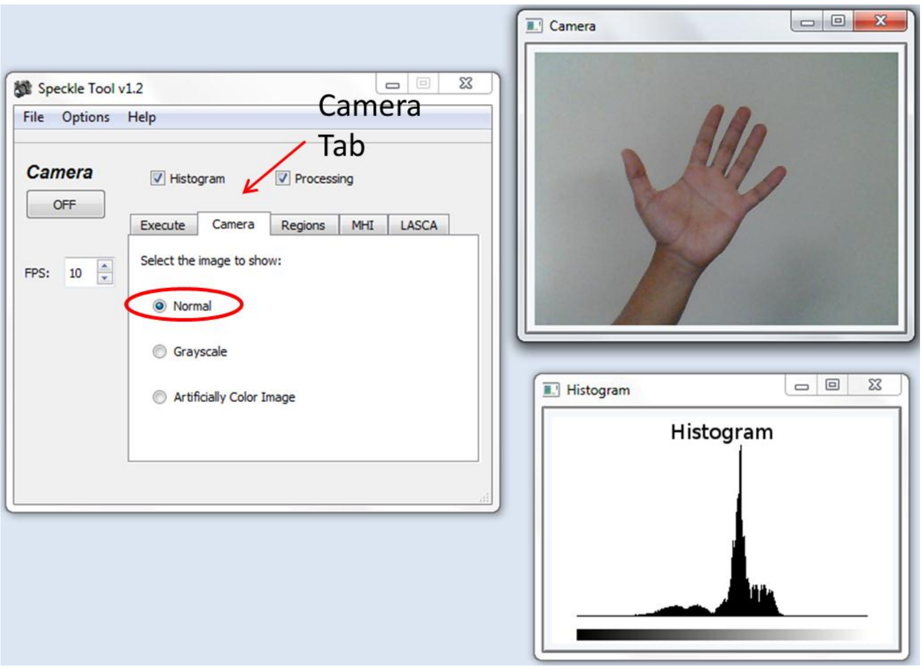
4. Camera Tab

The simplest action executed by the program is show the camera image without any processing, only with the information of the histogram, if it is active. This is also the default action (preset) program. When you open the program, without changing any settings, and turn on the camera, the camera's live image will appear in a new window called "Camera", and the histogram is displayed in another window called "Histogram".

At this tab there are three options: Normal, Grayscale and Artificially Color Image.

4.1 Normal Image

In Normal option, the camera image is showed without any image processing.



☑ Histogram ☑ Processing

FPS: 10

Select the image to show:

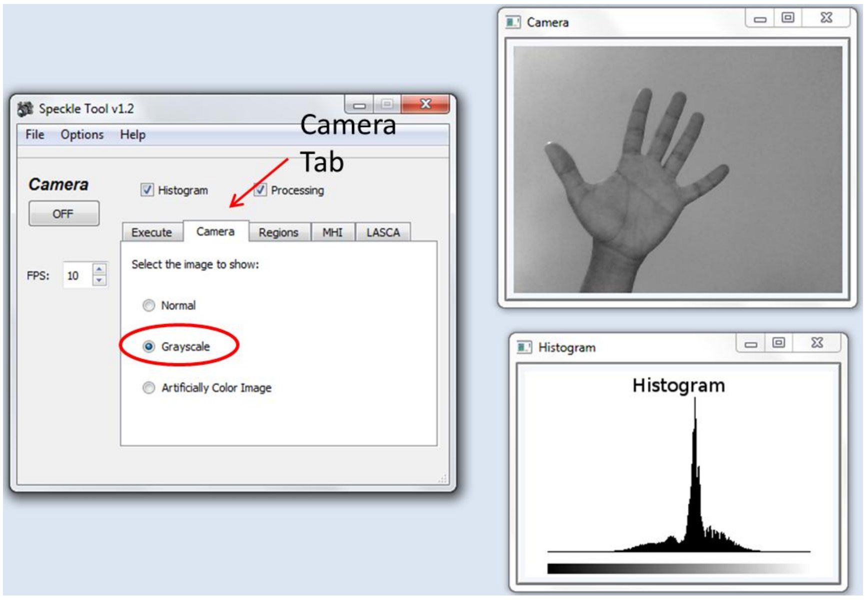
● Normal

○ Grayscale

○ Artificially Color Image

4.2 Grayscale Image

By selecting the option "Grayscale" in the "Camera" tab, the camera image passes through a color processing. The image is transformed into 8-bit image, gray levels. Effect very often used to process images speckle. So the image is showed in grayscale and no more in normal color.



Speckle Tool v1.2

File Options Help

Camera

OFF

☑ Histogram ☑ Processing

Execute | Camera | Regions | MHI | LASCA

Select the image to show:

○ Normal

● Grayscale

○ Artificially Color Image

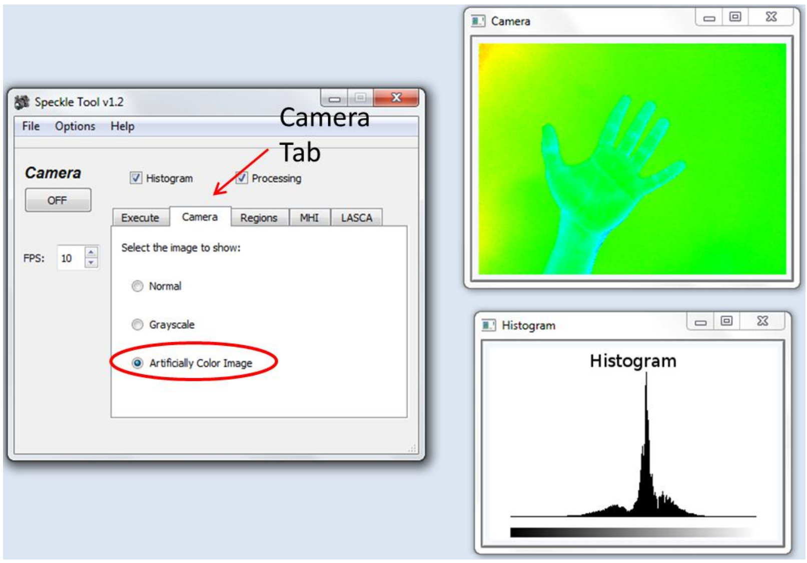
FPS: 10

4.3 Artificially Color Image

The camera image is transformed into a grayscale image and is used on this gray image a model conversion (Annex II) to apply a color palette default (Annex I). In this way the image is replaced by artificial colors that highlight saturated and dark areas of an image.

This effect helps the researcher identify saturated areas and/or underexposed areas of an image.

Once seen as the image histogram is too saturated or very underexposure, in other words, with there are many values near 255 or 0, choose this option and see which areas are next to the color red (color represents the maximum gray level) or blue (color representing the minimum gray level) and evaluate whether the area is in an important position of the image.



Speckle Tool v1.2

File Options Help

Camera

OFF

☑ Histogram ☑ Processing

Execute | Camera | Regions | MHI | LASCA

Select the image to show:

○ Normal

○ Grayscale

● Artificially Color Image

FPS: 10

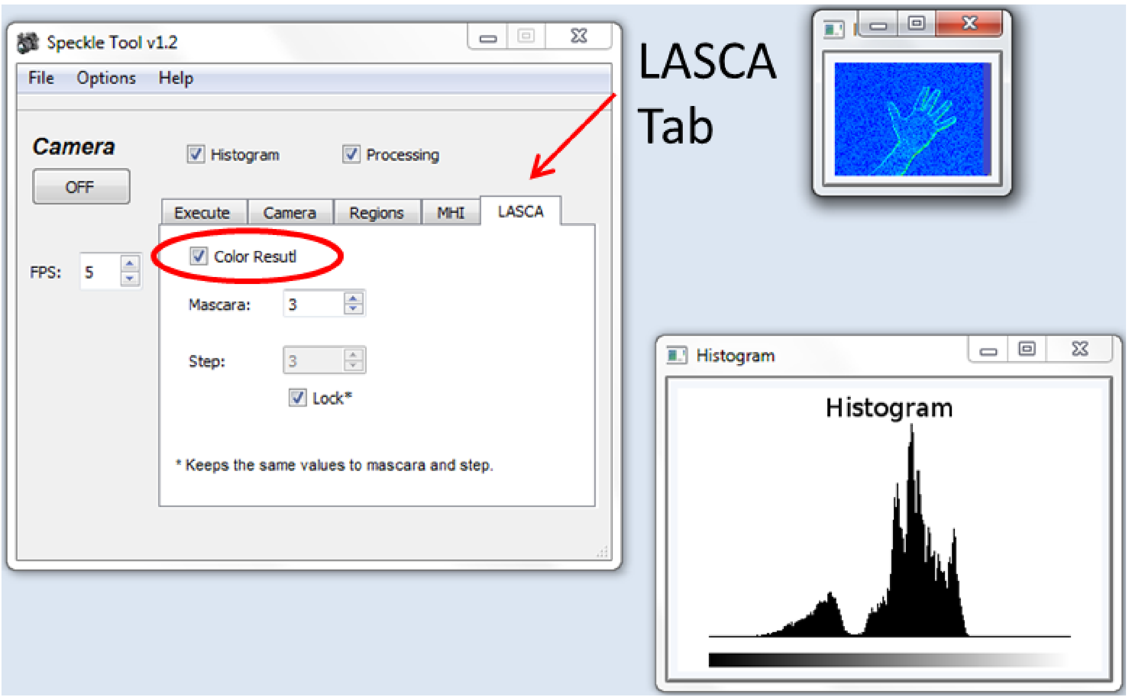
5. LASCA Tab

At this image processing option is executed the Laser Speckle Contrast Analysis (LASCA) algorithm, developed and published by Briers JD, known in the academic area of Dynamic Speckle Laser.

5.1. Color Result

The LASCA algorithm results in a grayscale image. This program has the option of artificially coloring the result of LASCA before displaying it, using the same conversion model presented in Annex II.

To see the LASCA color result must just check the “Color Result” check box.



FPS: 5

☑ Histogram ☑ Processing

☑ Color Resutl

Mascara: 3

Step: 3

☑ Lock*

* Keeps the same values to mascara and step.

5.2. Mask

This option adjusts the size of the mask used in the LASCA algorithm. The final image size is inversely proportional to the size of the mask. The original Briers JD algorithm uses the mask value equal to the step value.

According Briers JD 1996, a good mask size being used is five or seven. Less than 5 does not guarantee reliability to the result of processing and greater than 7 decreases much the resolution of the result image.

This parameter can assume the values: 3, 5, 7 and 9.

5.3. Step

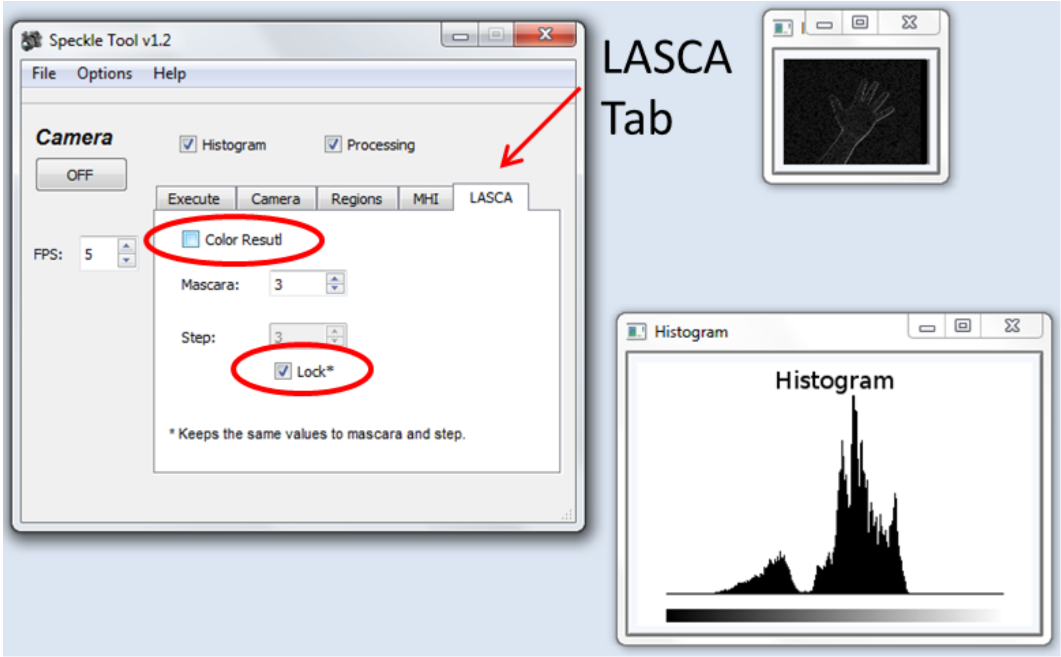
Step is the name given to the movement of the mask in the algorithm. The original algorithm proposed by Briers has the same step and the mask values. But change the step value generates some interesting results in certain cases.

Actually it is the step that changes the resolution of the result image, so the final image size is inversely proportional to the step size and not to the mask size.

Observation:

To execute the Classic LASCA processing must simply check the check box labeled "Lock". When this check box is checked, the parameter "Step" is disabled and is set with the same value of the parameter "Mask".

The possible values for this parameter are from 1 up to 9, only integer numbers.



LASCA Tab

FPS: 5

☑ Histogram

☑ Processing

□ Color Resutl

Mascara: 3

Step: 3

☑ Lock*

* Keeps the same values to mascara and step.

6. MHI Tab

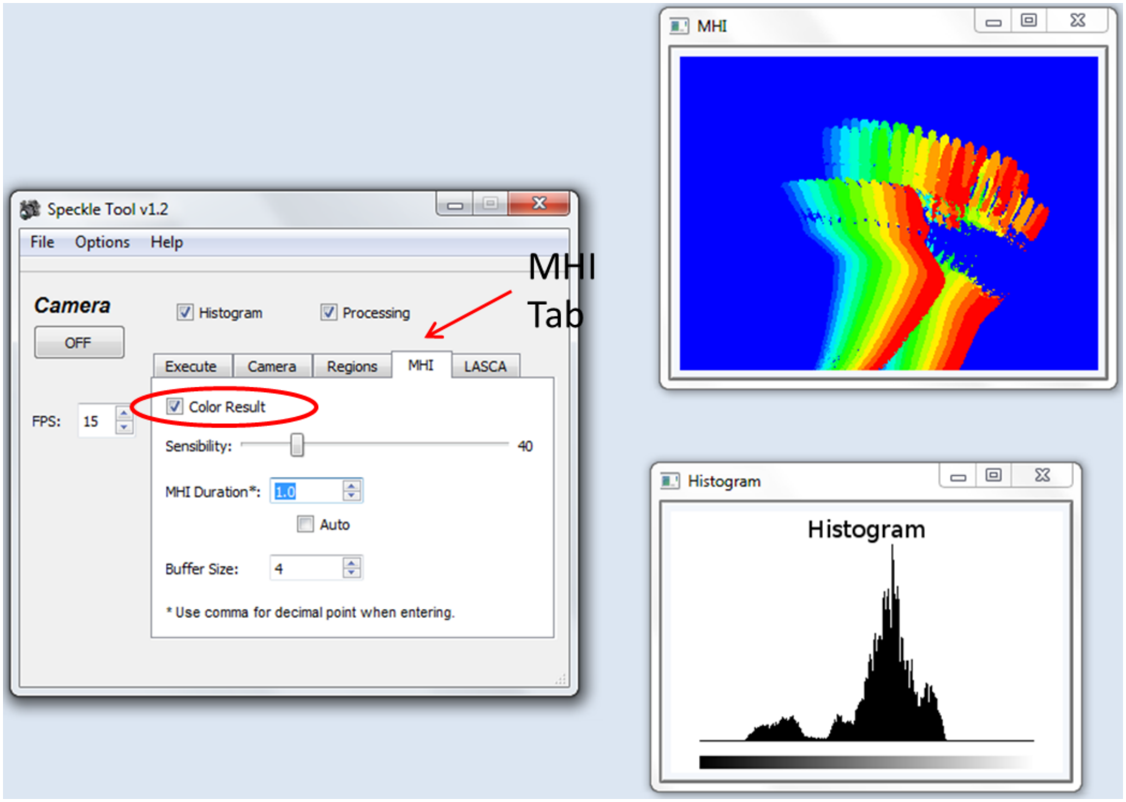
This option executes the Motion History Image (MHI) algorithm, developed and published by Davis JW, on collected camera images.

To get a better view of MHI is advised a high Frames Per Second (FPS) rate.

6.1. Color Result

MHI algorithm results in a grayscale image. This program has the option of artificially coloring the result of MHI before displaying it, using the same conversion model in Annex II.

To see the MHI color result must just check the “Color Result” check box.



Camera: OFF

☑ Histogram ☑ Processing

FPS: 15

☑ Color Result

Sensibility: 40

MHI Duration*: 1.0

□ Auto

Buffer Size: 4

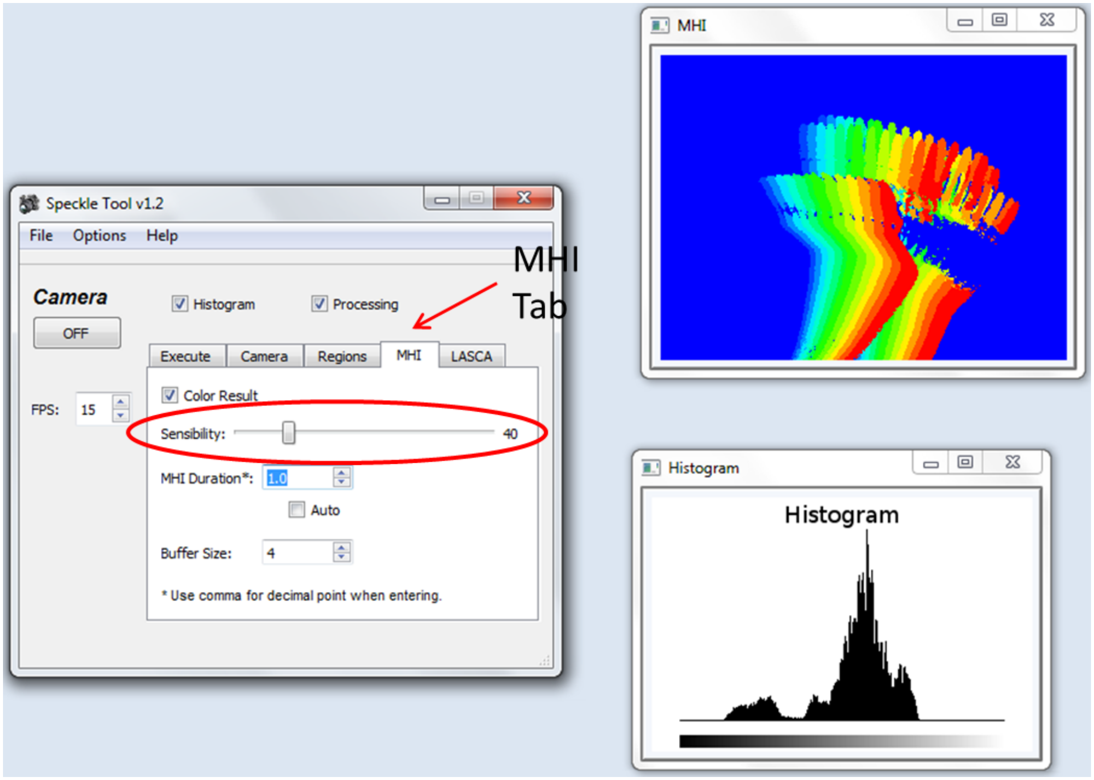
* Use comma for decimal point when entering.

6.2. Threshold

Controls Threshold parameter value used to modify the motion sensibility of the MHI algorithm.

How lower the threshold value is more sensitive is the algorithm and more discrete movements are captured. With a very high threshold value only abrupt movements will be captured by the algorithm.

This parameter can take values from 0 up to 255, only integer numbers, which represent the gray levels.



FPS: 15

☑ Histogram ☑ Processing

☑ Color Result

Sensibility: 40

MHI Duration*: 1.0

□ Auto

Buffer Size: 4

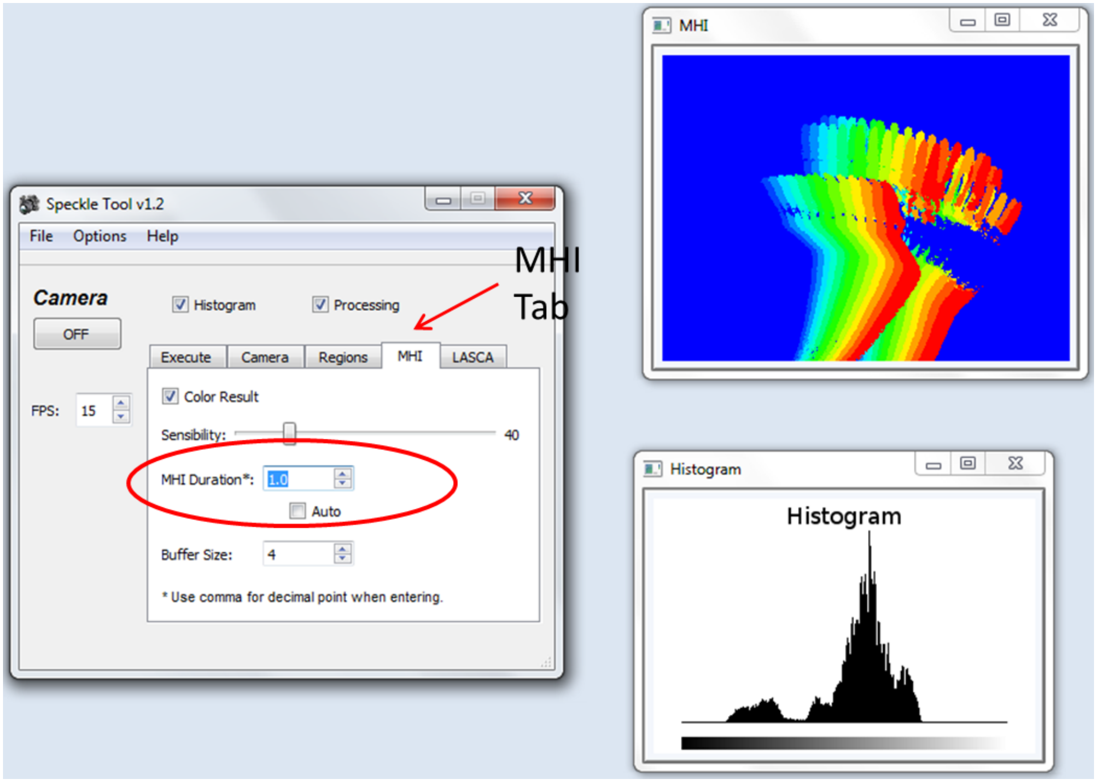
* Use comma for decimal point when entering.

6.3. MHI Duration

It controls the MHI duration Parameter utilized on the algorithm, representing the lifetime of a captured image in the process. At the end of this time, the image is no longer part of the set of historical images used in the processing.

The check box "Auto", automatically calculates the MHI Duration value from the values from the "Buffer size" and "FPS". Given that the lifetime of a frame is the "Buffer Size" value divided by "FPS" value. It is the conventional way to execute this processing.

This parameter assumes values form 0.1 up to 128, with intervals of 0.1. If you will type in this field, use comma for decimal numbers.



☑ Histogram ☑ Processing

FPS: 15

☑ Color Result

Sensibility: 40

MHI Duration*: 1.0

□ Auto

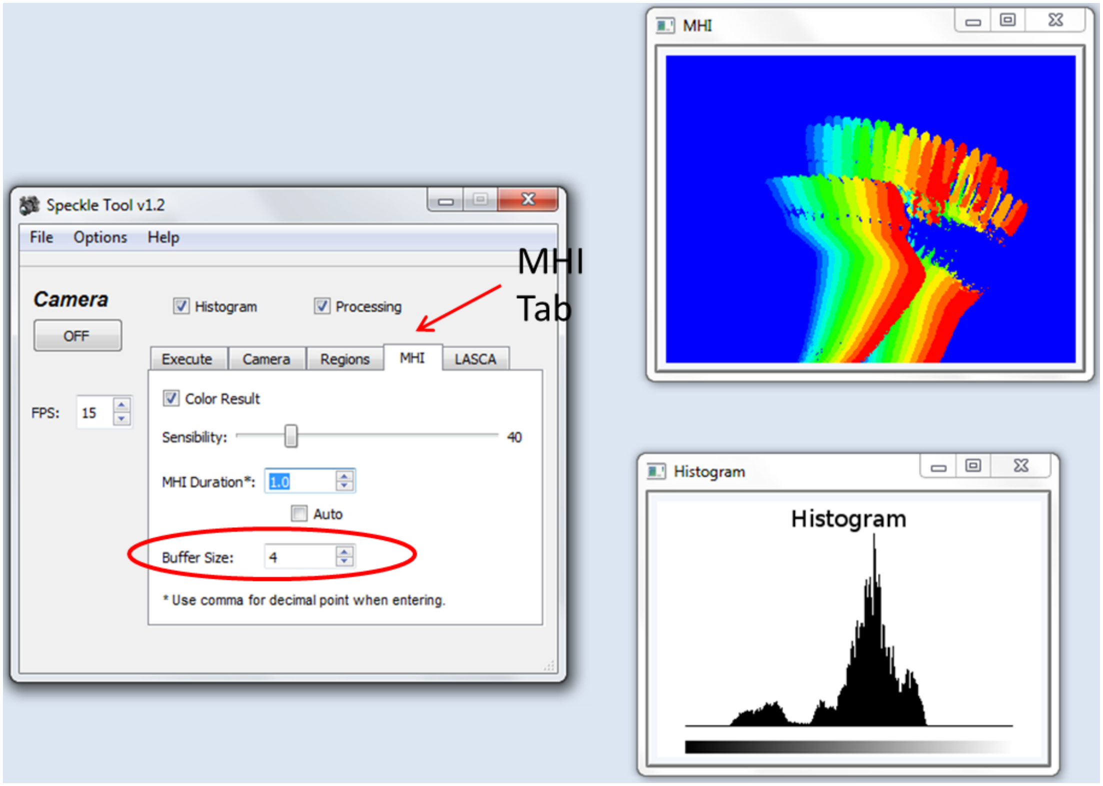
Buffer Size: 4

* Use comma for decimal point when entering.

6.4. Buffer Size

It controls the buffer size parameter of the MHI algorithm, i.e., how many images are used to execute the algorithm processing.

This parameter can assume values from 2 up to 128, only integer numbers.



☑ Histogram ☑ Processing

FPS: 15

☑ Color Result

Sensibility: 40

MHI Duration*: 1.0

□ Auto

Buffer Size: 4

* Use comma for decimal point when entering.

7. Regions Tab

This option creates a line that identifies the area with the greatest activity, delimiting a region in the image. This way is possible to know which area is the most active.

This option realizes the combination of algorithms and techniques for image processing in 6 steps, so it has a very high computational effort. It is recommended not use a frame per second (FPS) rate high, to not overload the system.

How this procedure can use the algorithms LASCA and MHI, the parameters of these algorithms must be set in their proper tabs.

7.1. Algorithms

There is the option to use or not separately two algorithms to execute this method: LASCA and MHI.

To enable these algorithms it is necessary to check their check box in the tab "Regions”.

In step 1, if the option to use the MHI algorithm is selected, the method will execute it on the camera images and the gray result image will be sent to step 2, otherwise the gray camera image will be sent to step 2.

In step 2, if the option to use the LASCA algorithm is checked, the LASCA will be executed on the image coming from step one and the gray result image will be sent to step 3, otherwise the image coming from step one will be sent directly to step 3 without any processing.

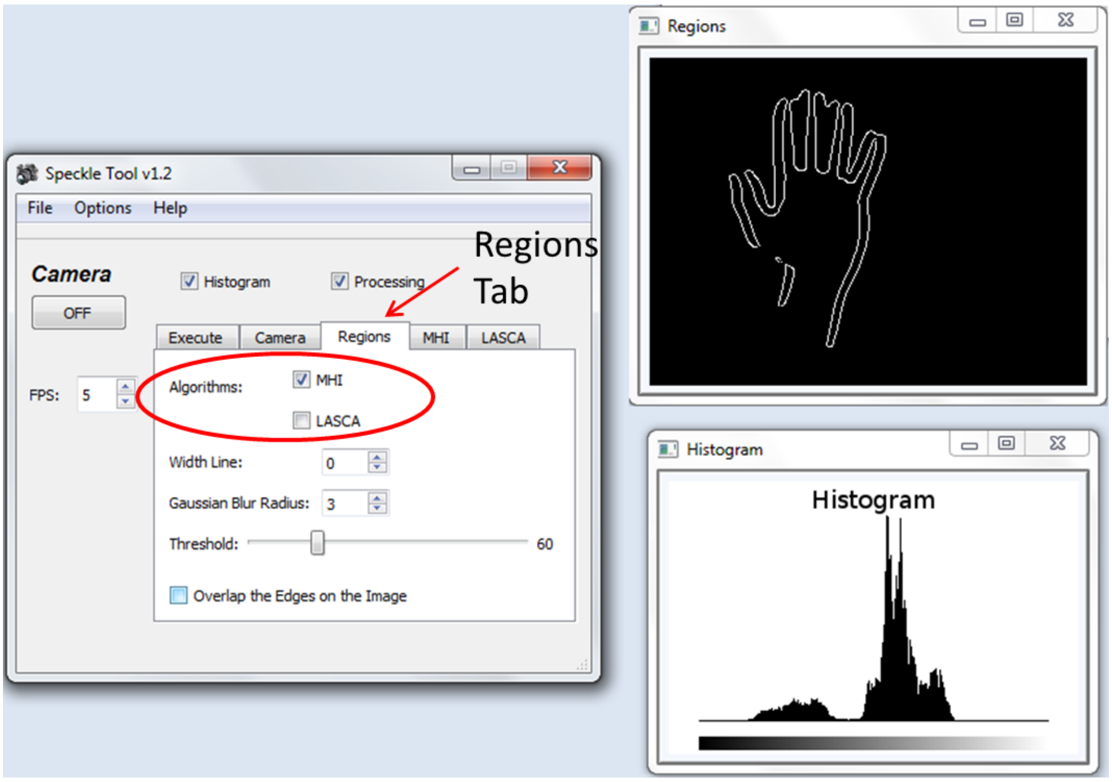
In step 3 is executed the Gaussian blur on the image coming from the step 2. This step is very important because the speckle grains will join to form a dense mass that can be identified as regions. After this process the blur result image will be sent to the step 4.

In step 4, the binary threshold is applied on the image received from the step 3. After these processing the result image will be sent to step 5.

In step 5, is applied the edge detection algorithm on the image received from step 4 and the result image will be sent to step 6.

In step 6, if the check box “Overlap the Edges on the image” is checked, the contour image received form the step 5 is applied on the camera image and the result is showed, otherwise the image received from the step 5 is showed as a final result.

All the steps above are executed sequentially and in real time, without the user realize about the execution and result of each one separately, only the final result is showed.



☑ Histogram ☑ Processing

FPS: 5

Algorithms: ☑ MHI ☐ LASCA

Width Line: 0

Gaussian Blur Radius: 3

Threshold: 60

☐ Overlap the Edges on the Image

7.2. Gaussian Blur Radius

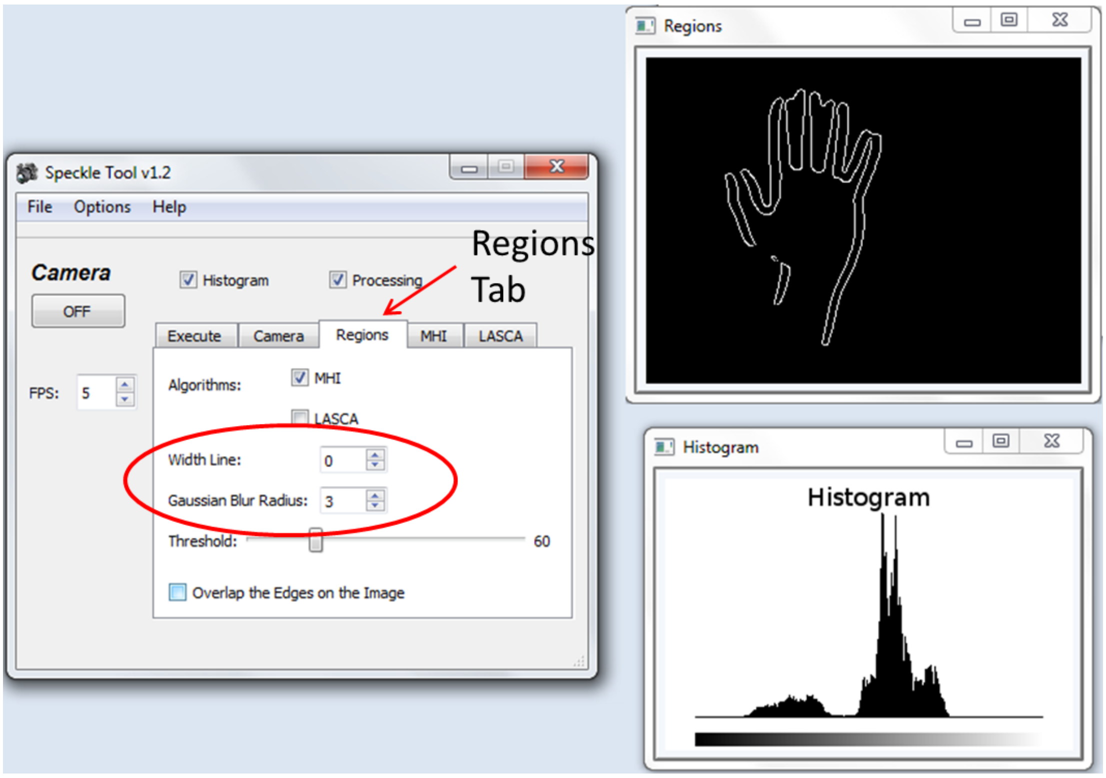
This option sets the parameter value used in step 3 of the Regions method, the radius of the Gaussian blur. The larger the radius, the denser the area gets and the more blurred the speckle image will be.

The values that can be assumed by this parameter are 1 up to 4 (integer numbers only).

7.3. Width Line

Changing this parameter the line width seen in the Regions method can be changed. If there is any trouble seeing or want to highlight the line must just increase the value of this parameter.

Observation: When the LASCA algorithm is used in regions method and the value of the parameter step in the LASCA tab is high, by implementation issues, the width line will increase naturally, without change the value of width line parameter.



Speckle Tool v1.2

☑ Histogram ☑ Processing

FPS: 5

Algorithms: ☑ MHI

☐ LASCA

Width Line: 0

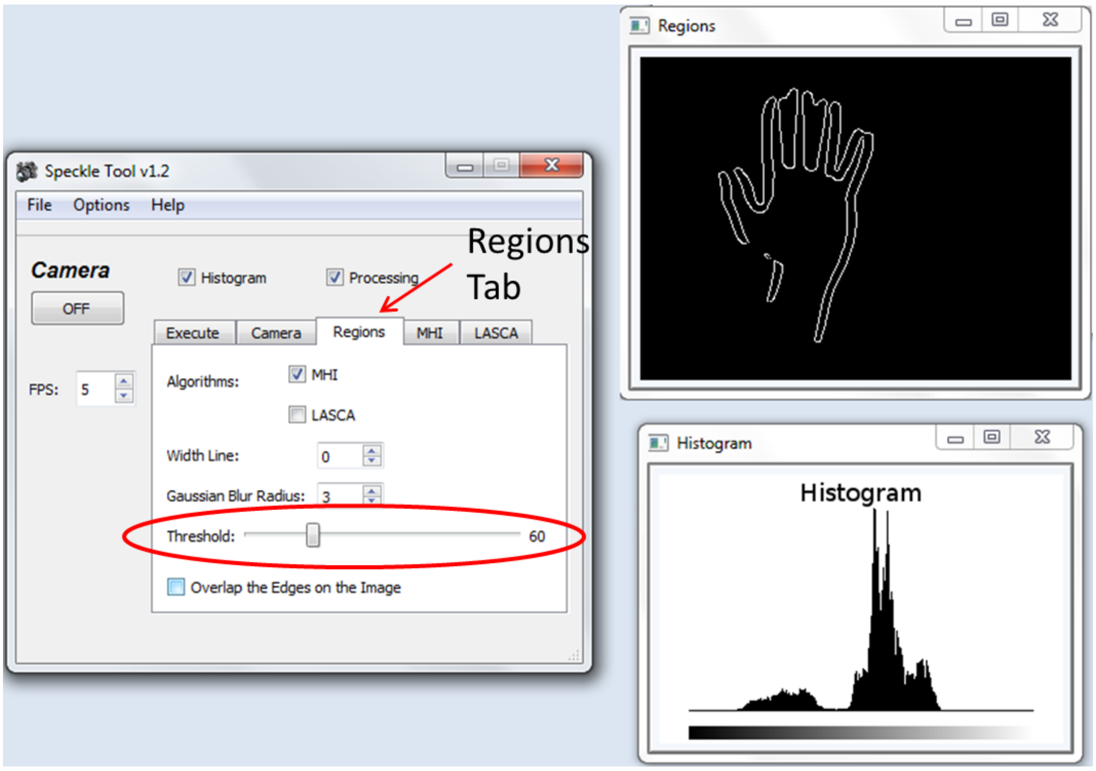
Gaussian Blur Radius: 3

Threshold: 60

☐ Overlap the Edges on the Image

7.4. Threshold

This control is designed to control the parameter of step 4. The assumed values range from 0 up to 255 (integer numbers only).



Speckle Tool v1.2

☑ Histogram ☑ Processing

FPS: 5

Algorithms: ☑ MHI

☐ LASCA

Width Line: 0

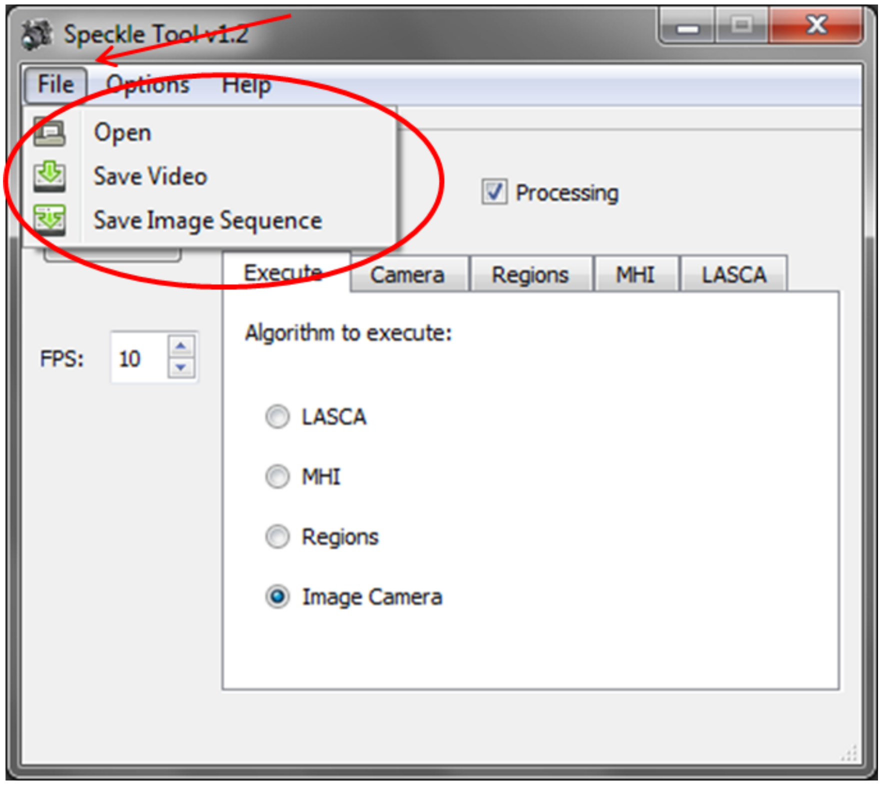
Gaussian Blur Radius: 3

Threshold: 60

☐ Overlap the Edges on the Image

8. Extra Functions

The program has some features to help the researchers as save image sequence, open video files, in format “.avi”, and save videos, in format “.avi” as well. These functions can be found in the menu bar, the menu “File”.



8.1. Open Video File

To open a video file, open the menu "File" in the menu bar and choose the option "Open." A file selection window, compatible with your operational system, will be opened. There you can choose only video files with avi extension (". avi"). Select the desired file and click Open.

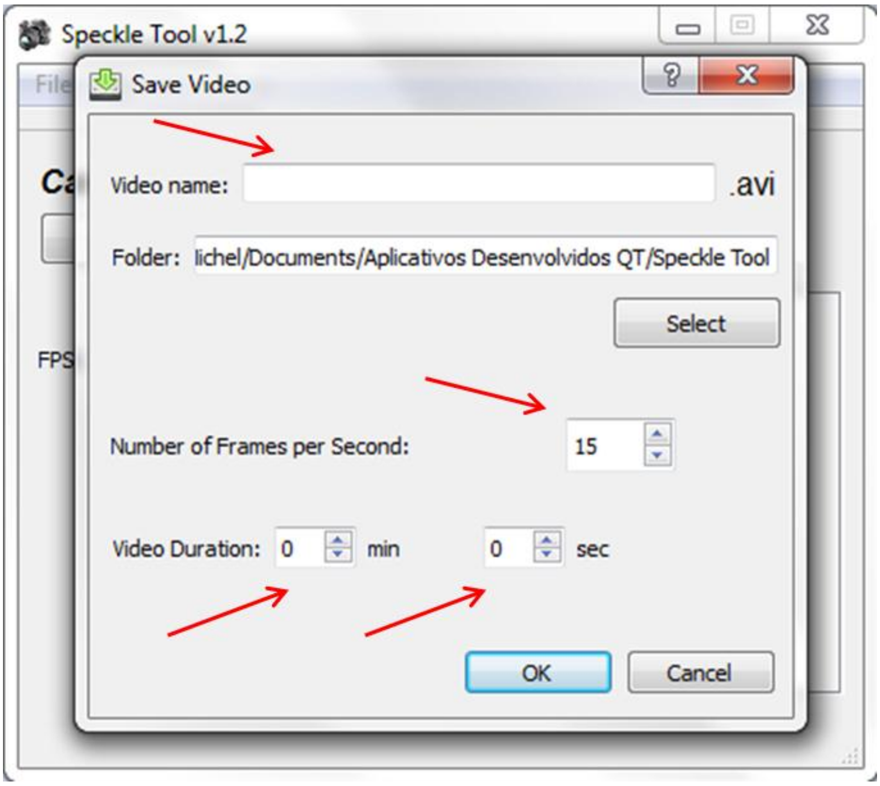
The video will be executed and the image processing will be executed on the video images and no more on the camera images.

At the end of the executing of video will be displayed the following message in the status bar for 5 seconds: "Finished executing the video.".

To open compressed videos like DIVX and XVID is necessary to have the appropriate codecs and OpenCV 2.0 or higher installed in your computer.

8.2. Save Video

To save video from the camera to a avi video file, open the menu “File”, in the menu bar, and click in “Save Video”.


Speckle Tool v1.2
File
Ca
FPS
Save Video
Video name:
.avi
Folder: lichel/Documents/Aplicativos Desenvolvidos QT/Speckle Tool
Select
Number of Frames per Second: 15
Video Duration: 0 min 0 sec
OK
Cancel

Set a name for the file. It is not necessary type the extension ".avi" at the end of the file name to be saved.

Select the folder to save the video. This can be typed or selected via selection window of the operational system by clicking the button "Select" just below the field “Folder”.

Set the number of "frames per second" rate that the video will be saved. Then select the duration of the video. **Always remember** to set the Video Duration**.**

Check all the fields and if you are sure about this, click the button “OK”.

While the recording processing the message "Recording video ..." will be displayed on the status bar of the program main window. When it finish the message "Video saved!" will be displayed for 5 seconds on the status bar.

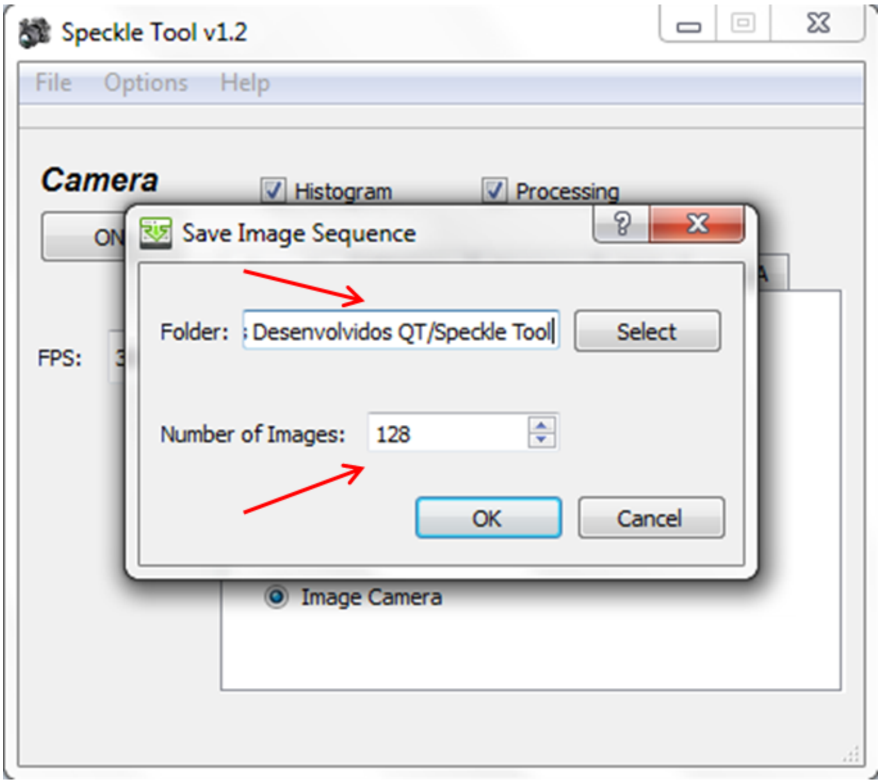
The saved video consists of the real images collected by the camera. You cannot make a video directly with result images of some processing by this software. To do it that collect a sequence of images and use a an auxiliary program to build a video from the sequence images.

If the camera is on, and the histogram showing the images, the image window and the histogram window will be closed and the camera restarted to start save the video. During the time of recording video ***images are not being displayed on the screen****.*

The video is saved using DIVX compression, so you must have the appropriate codecs and OpenCV 2.0 or higher installed in your computer.

8.3. Save Image Sequence

To save the results of the image processing, must just open the menu "File" in the menu bar and click in "Save Image Sequence". Thus the images being displayed on the screen will be saved on computer disk.



Folder: ; Desenvolvidos QT/Speckle Tool

Number of Images: 128

Enter or select a folder to save the images and set the number of images you want to save. After that just click “OK”.

The number of images should be a value between 1 and 300 (only integer numbers).

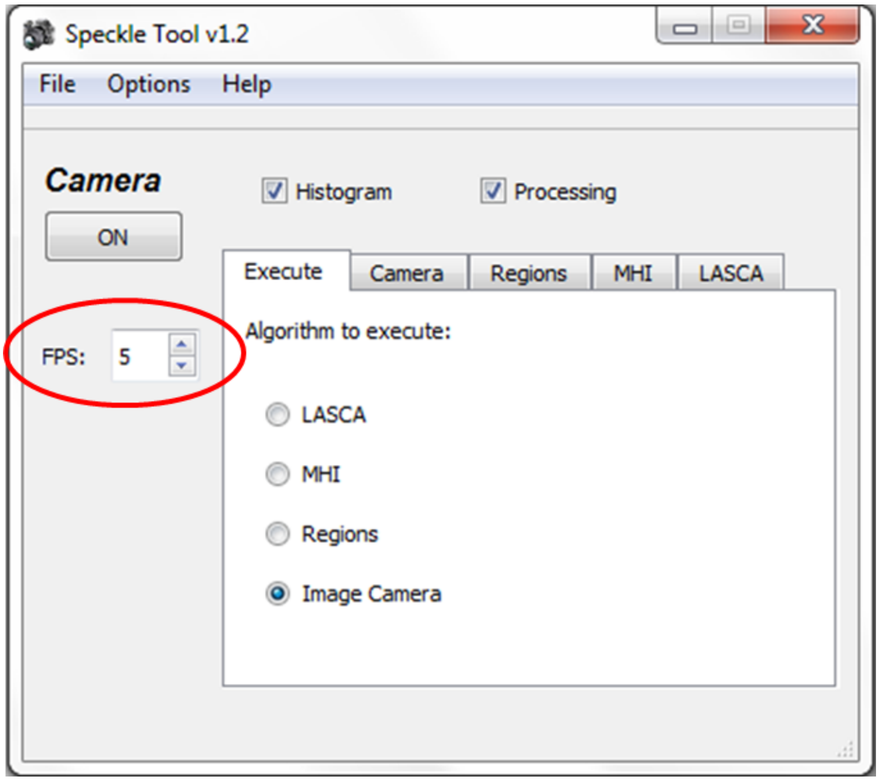
While pictures are being saved, the message "Saving Images, wait ..." will appear in the status bar of the program main window. At the end of saving the images, the message "Images saved successfully" will be displayed for 5 seconds.

If the camera is on, and the histogram and the images windows are being showed, those windows will be close and the camera restarted to start save the image sequence. During the time that the images are being saved, those images will not be displayed on the screen.

9. Configuration

9.1. FPS

This field is visible in the main window of the program throughout the execution of that. By it is possible to configure how many images per second will be collected by the camera (Frames Per Second - FPS).



FPS: 5

Algorithm to execute:

○ LASCA

○ MHI

○ Regions

● Image Camera

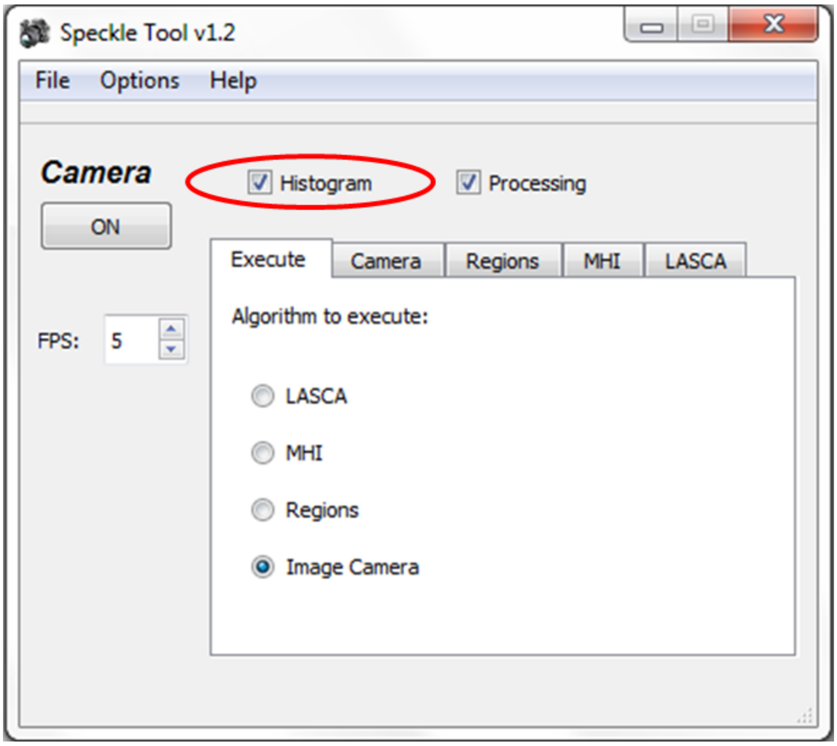
How lower is the value of this parameter less pictures will be collected by the camera and you will feel a slow motion video, with the image freezing. How higher is this value faster are the video.

The values taken by this parameter are 1 up to 30 (integer numbers only).

Make sure your camera is able to capture the number of images per second that you set the FPS parameter in this program.

9.2. Histogram

When the camera is on or is running a video from file is displayed the histogram of the image that are being processed. To disable this functionality must simply uncheck the check box "Histogram". Thus only the final image processing is displayed on the screen.



☑ Histogram ☑ Processing

FPS: 5

Algorithm to execute:

○ LASCA

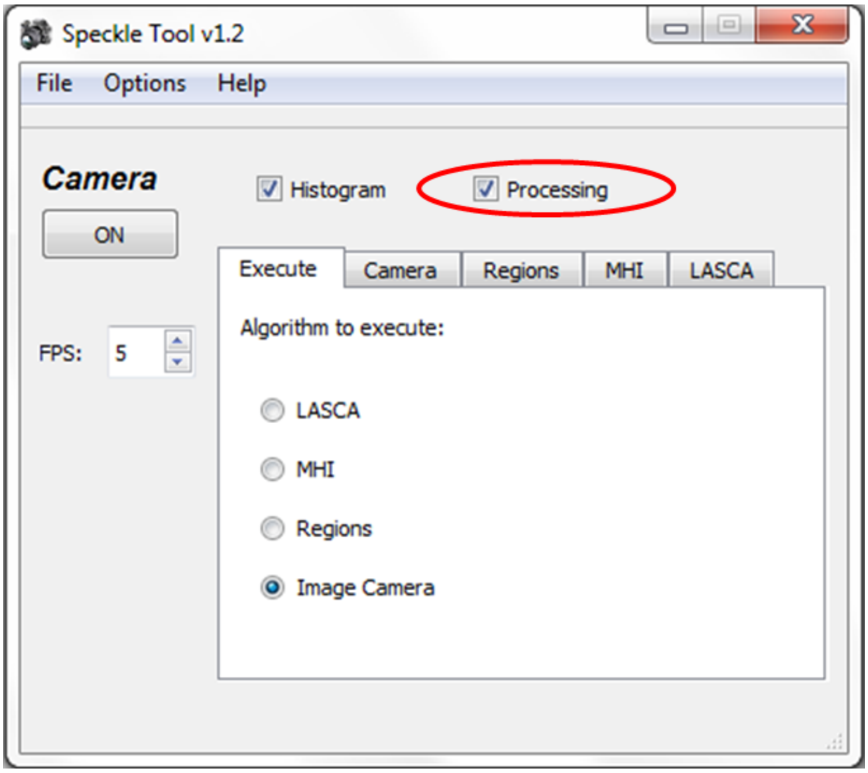
○ MHI

○ Regions

● Image Camera

9.3. Processing

If this check box is checked, image processing and result display are enable, so the image will be processed and showed. If it is unchecked, the images are not processed neither showed.



☑ Histogram ☑ Processing

FPS: 5

Algorithm to execute:

○ LASCA

○ MHI

○ Regions

● Image Camera

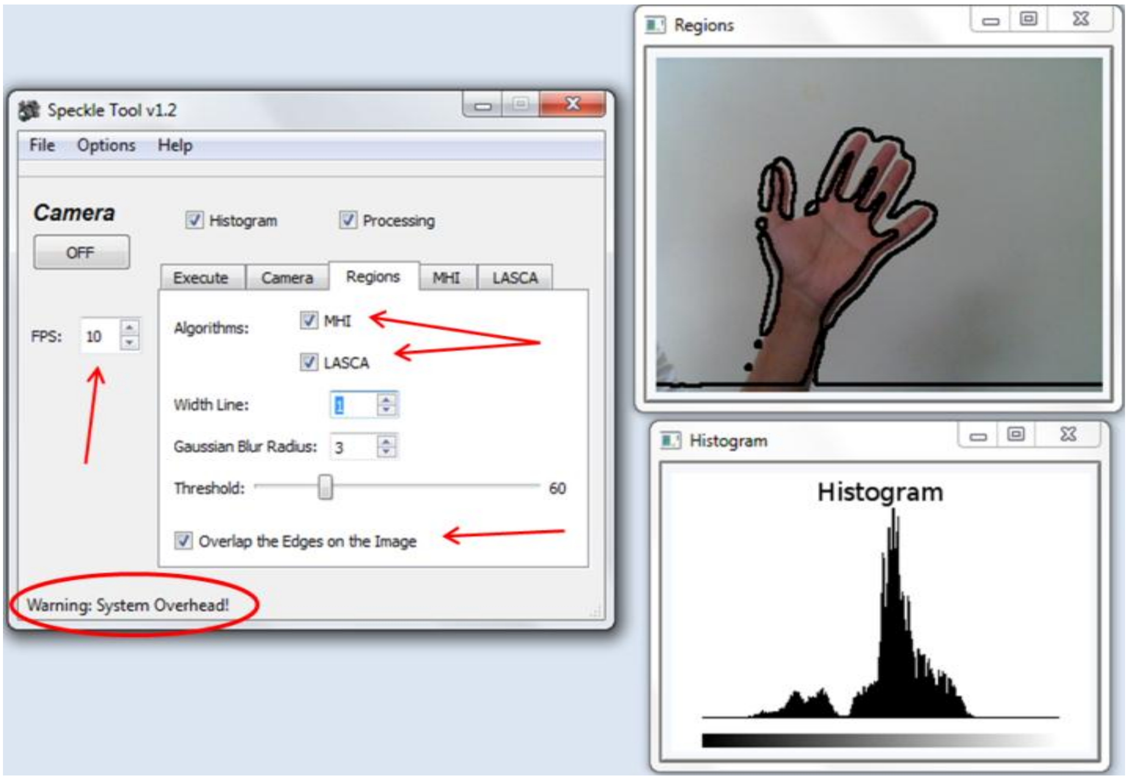
9.4. System Overhead Warning

When the system is overloaded, a warning message will appear in the status bar of the main program window. This warning indicates that the computer does not have the performance necessary to achieve the processing in the set rate Frames Per Second.

This message appears when the time taken to process the images is very high, and the rate of Frames Per Second set is not being obeyed. For example, if the FPS value is 20, which means 20 images will be captured and processed per second, so the time that can be spent with each image processing is 1/20 = 0,05 seconds.

Example: Let's suppose to execute the LASCA algorithm with mask value 3, the computer spend 0,75 seconds per image and the FPS value set is 20. The maximum time that the computer has to execute and show the LASCA result is 0,05, as view above. But the computer is spending 0,75 seconds, time higher than the permitted, so the computer is not able to execute this image processing in this frame rate.

Tip: If the message is displayed, set the FPS value lower to ensure the reliability of the results.



☑ Histogram ☑ Processing

FPS: 10

Algorithms: ☑ MHI ☑ LASCA

Width Line: 1

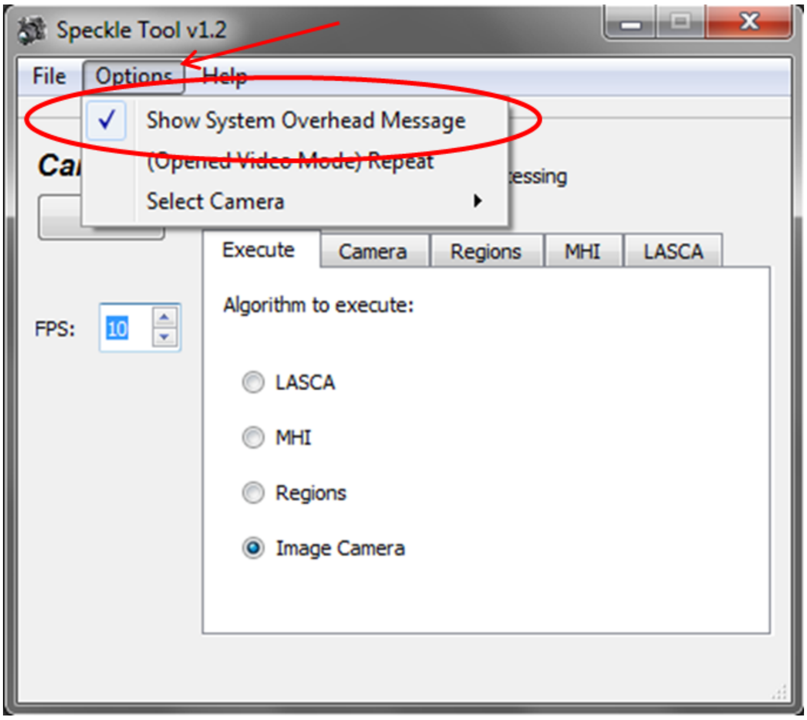
Gaussian Blur Radius: 3

Threshold: 60

☑ Overlap the Edges on the Image

Warning: System Overhead!

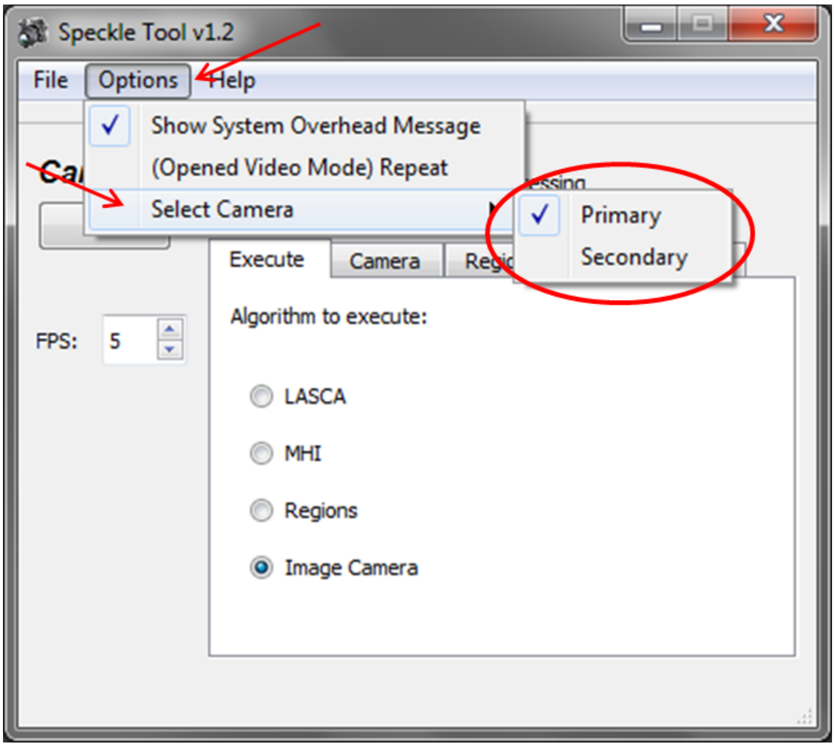
This function can be disabled to not show the message in the status bar when the computer is overloaded. To do this, open the menu "Options" in the menu bar and uncheck "Show Message Overflow system". **Important:** it just won't show the message, the system continues overloaded. It is not recommended uncheck this option.



9.5. Select Camera

When the computer is plugged with two USB cameras, the software give you the option to choose between them. **Important: this program only works with USB cameras.**

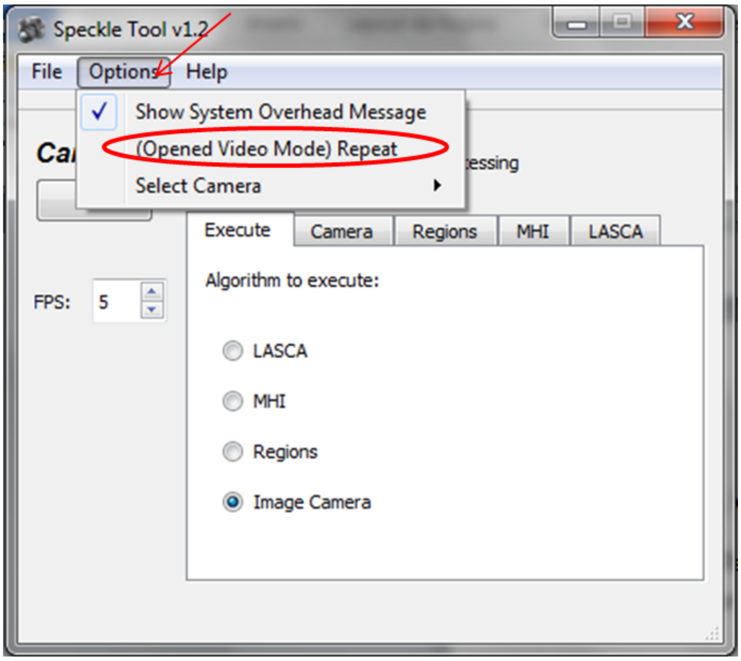
To select the active camera open the menu "Options" in the menu bar, go to option "Select Camera" and select the desired camera.



9.6. Repeat Mode of Opened Video

To set the video in infinite repeat mode must open the menu "Options" and check the option “(Opened Video Mode) Repeat”.

With this option enabled the video will be executed in repeat mode infinitely. To stop the video you need open the menu "Options", uncheck this option and then wait the video to be finalized.



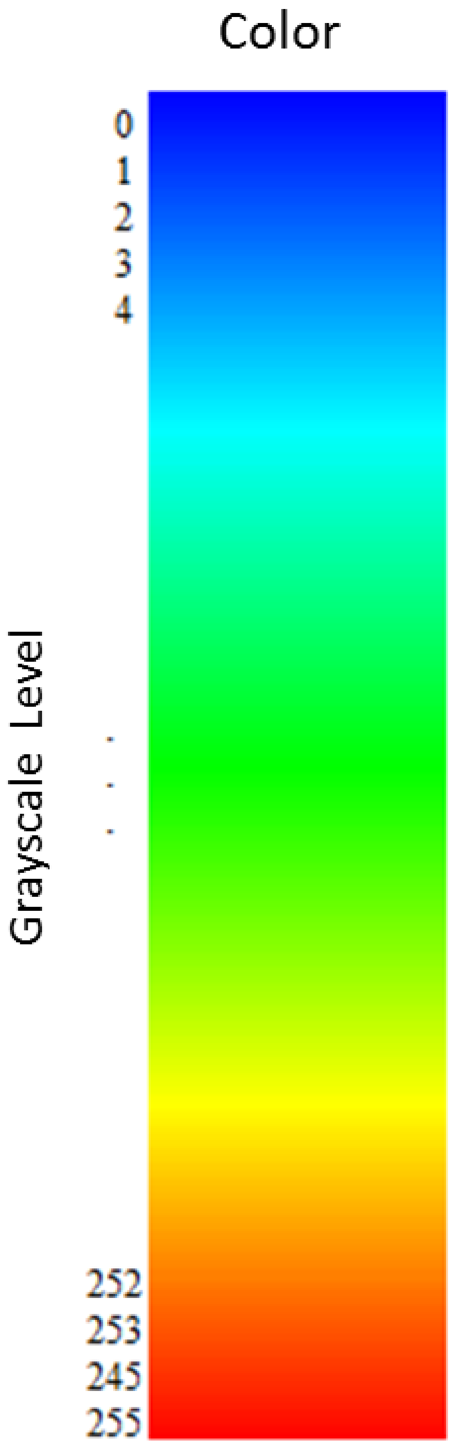
Notes:

The option “Repeat Mode of Opened Video” is only available from version 1.2 onwards.

From the version 1.1 the program is only available in English.

Annex I

Color palette used in this software:



Annex II

Color conversion model used:

Conversion Model

Being Gray_Level equal to the intensity value of the pixel P at position (x,y) of image I.

$$I_{P(x,y)} = Gray\_Level$$

For each pixel of the Image I do:

```
if(Gray_Level < 64)
{
    R(x,y) = 0;
    G(x,y) = (Gray_Level * 4);
    B(x,y) = 255;
} else
{
    if(Gray_Level < 128)
    {
        R(x,y) = 0;
        G(x,y) = 255;
        B(x,y) = (255 - (Gray_Level - 64) * 4);
    } else
    {
        if(Gray_Level < 192)
        {
            R(x,y) = ((Gray_Level - 128) * 4);
            G(x,y) = 255;
            B(x,y) = 0;
        } else {
            R(x,y) = 255;
            G(x,y) = (255 - (Gray_Level - 192) * 4);
            B(x,y) = 0;
        }
}
```

Where R(x,y) means red channel value of the pixel at position (x,y), G(x,y) means green channel value of the pixel at position (x,y) and B(x,y) means the value of the pixel at position (x,y) of the final image. Each pixel is formatted by the (R,G, B) channels.